IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

MARK A. BAKKE et al.

Serial No.: 09/373,795

Filed: August 13, 1999

For: DISTRIBUTED FILE DATA LOCATION

Attorney Docket No.: 98-127-NSC/STK98127PUS

Group Art Unit: 2177

Examiner: S. Channavajjala

REPLY BRIEF UNDER 37 C.F.R. § 1.193(b)(1)

Box AF
Commissioner for Patents
United States Patent and Trademark Office
Washington, D.C. 20231

Sir:

Please accept this is a brief in reply to the Examiner's Answer mailed October 17, 2002. This brief supports the Supplemental Appeal Brief mailed September 26, 2002, and should not be considered without this Appeal Brief.

In the Examiner's Answer, the Examiner provided Grounds of Rejection which appear to be identical to those used in the final Office Action mailed February 8, 2002. Appellants believe that these grounds have been fully refuted in the Appeal Brief and, to as great an extent as possible, will not be simply repeated in this Reply Brief. Instead, Appellants will address the Examiner's responses to Appellants' arguments where the Examiner appears to offer additional support.

CERTIFICATE OF MAILING UNDER 37 C.F.R. § 1.8

I hereby certify that this paper, including all enclosures referred to herein, is being deposited with the United States Postal Service as first-class mail, postage pre-paid, in an envelope addressed to: Box AF, Commissioner for Patents, United States Patent and Trademark Office, Washington, D.C. 20231 on:

December 13, 2002	Mark D. Chuey, Ph.D.	
Date of Deposit	Name of Person Signing	Signature

1. Story Does Not Teach or Suggest Separate Location Servers and Name Servers

Independent claims 1, 11 and 16 each provide for a name server which is "separate" or "physically separate" from a location server. The Examiner asserts that U.S. Patent No. 6,081,807 to Story *et al.* (Story) teaches such separate servers. Appellants disagree.

Appellants' invention locates data within a heterogenous distributed file system. The prior art, including Story, describe a three element system for locating data: clients, servers and data storage. This three element system is illustrated in Appellants' Figure 1 and described on page 5, lines 8-23, as follows:

> Referring to Figure 1, a schematic diagram illustrating a prior art client-server relationship is shown. A file system, shown generally by 20, includes clients 22 accessing data held in files on one or more storage devices 24. Each storage device 24 is accessed through a server, one of which is indicated by 26. A typical example of file system 20 is the Unix-based Network File System (NFS). In NFS, client 22 wishing to access data first forwards the name of the file containing the data to server 26, as shown by 28. Server 26 returns a handle to the requested file as indicated by 30. Client 22 then forwards a data request with the received handle to server 26, as indicated by 32. Server 26 requests the data from storage device 24, as indicated by 34. Storage device 24 returns the data to server 26, as indicated by 36, and server 26 forwards the data to client 22, as indicated by 38. Another typical example of file system 20 is embodied in the CIFS (Common Internet File System) found in the WINDOWS NT ® server by Microsoft Corp. Server 26 provides a file descriptor in response to a name provided by client 22. Client 22 uses the file descriptor to access data through a logical connection through server 26 to storage device 24. The file descriptor is valid only for the life of the connection.

This appears to be precisely the same arrangement disclosed in Story. Story's Figure 1 illustrates "network client" 102, "network server" 106 and remote "data storage device" 132. Client 102 is separate from server 106 as indicated by network 110 interconnecting the two.

The Examiner asserts that Story's NFS server 122 is Appellants' location server and that Story's name server 130 is Appellants' name server. Both are illustrated by Story's

Figure 1 as being within – part of – network server 106. The Examiner's sole support that Story teaches "separate" or "physically separate" name servers and location servers is from page 21, as follows:

> It is clear from the Story's fig 1 that both NFS server, element 122 and name server element 130 are not only different servers but also physically different, further these tow servers are connected to network through various components as detailed in fig 1.

The Examiner's argument can be summarized thus: Since Figure 1 shows two different boxes, these must be "separate" or "physically separate" servers. This conclusion belies both Story's disclosure and the use of these terms to describe Appellants' invention.

Story discloses that various boxes shown within network server 106 are software elements running on server 106. This arrangement is described in column 4, lines 4-20, which is reproduced as follows (emphasis added, bracketed material added):

> *In network server 106, NFS server 122* is linked, via an interface 126, to a disk file system, such as an OSS (Open Systems Services) file system 124 developed by and available from Tandem Computers Incorporated, Cupertino, Calif. The phrase "file system" has several distinct meanings in computer science. As used in the application, unless otherwise qualified, a "file system" refers to a body of software designed to store, organize, protect, and retrieve data using some storage medium such as disk. OSS file system 124 is linked to a disk process 128 and a *name server 130* which is also linked to disk process 128. Disk process 128 is linked to a data storage device 132 and a file manager 134. It should be understood that, in the embodiment of FIG. 1, *elements* 112, 114, 116, 118, **[in network client 102]** *122*, 124, 126, 128, *130* and 134 **[in network server 106]** *are implemented as software programs stored in memory and executed by one or more respective processors* (not shown).

Story's NFS server 122 and name server 130 are software modules running on the same network server 106. They are not "separate" or "physically separate" as provided by Appellants.

Appellants show a separate client 52, name server 60, location server 66 and storage 54 in Figures 2, 5 and 6. Figure 2, for example, illustrates clients, name servers, location servers and data storage interconnected through networks 58, 64, 70, described as "local area networks (LANs), wide area networks (WANs), storage are networks, (SANs), and the like." (Pg. 6, ll. 29-30.) In this embodiment, the only access between any name server 60 and location server 66 is through at least one network.

Thus, Story neither teaches nor discloses "separate" or "physically separate" name servers and location servers.

<u>2. Story Does Not Teach or Suggest Appellants' Name Database, Location Database and Client as Provided in Claim 16</u>

Claim 16 provides a file system for storing data including storage devices, at least one location database, at least one name database and at least one client. Each location database includes a map between a file identifier for each file and location information for each copy of the file represented by the file identifier. Each name database includes a map between a file name and the file identifier referenced by the file name. Each client requests a file identifier corresponding to a requested file name, receives the file identifier mapped to the requested file name, requests location information corresponding to the received file identifier, receives location information mapped to the received file identifier, and accesses data using the location information. Thus, the client initiates mapping in both the location database and the name database and the client receives the results of both of these mappings.

The Examiner asserts that Appellants' client is taught by Story as "client(s) and server(s) through network such as detailed in fig 1, specifically at least one client, see fig 1, element 102, fig 2, network client, examiner interpreting client corresponds to Story's network client as detailed in fig 1, element 102." (Page 23.) Whatever this may mean, it in no manner describes the requesting and receiving role of Appellants' client as provided in claim 16. As is unambiguously clear from Story's Figure 2, the only information which flows from Story's network client to Story's network server is a read or write request (col. 5, ll. 22-23) and the

only information which returns to the network client from the network server is the read data or write status (col. 6, ll. 19-20).

Thus, Story neither teaches nor suggests Appellants' client as provided in claim 16.

3. Story Does Not Teach Appellants' Name Server and Location Server as Provided in Claim 11

Claim 11 provides a method for accessing a file referenced by a file name. The file name is sent to a name server. A file identifier corresponding to the file name is received from the name server. The file identifier is sent to a location server which is separate from the name server. File location information corresponding to the file identifier is received from the location server. The file is accessed using the location information. Embodiments of this concept are illustrated in Appellants' Figure 5, for a write operation, and Figure 6, for a read operation. In both cases, name server 60 receives a command for name lookup (110, 130) and returns a handle (112, 132). Location server 66 then receives a command containing the handle (114, 134) and returns one or more storage locations (116, 136).

Story fails to teach Appellants' name server, fails to teach Appellants' location server, and fails to teach a file identifier received from a name server and sent to a location server.

a. Story Does Not Teach Appellants' Name Server

Claim 11 provides a name server which receives a file name and returns a file identifier corresponding to the file name. The Examiner has identified Story's name server 130 as Appellants' name server, stating "firstly, name server is responsible for file name hierarchy and provides pathname, secondly, Story teaches for example file handle which contains information such as type of file, unique identifier or file ID as detailed in col 4, line 41-47." (Page 26.)

As to the first argument, whether or not Story's name server "is responsible for file name hierarchy and provides pathname" is irrelevant since this does not disclose receiving a file name and returning a file identifier. As to the second argument, the passage cited by the Examiner does not refer to Story's name server 130. The passage is reproduced as follows:

> The NFS LAN interface process dispatches work to a server process in NFS server 122, based on the contents of a "file handle," which is a parameter in the NFS request. A file handle contains such information as the type of file, the time of creation of the file, a unique identifier (file set ID) for the file set in which the file resides, a unique identifier (file ID) for the file within the file set, etc.

The interface process, in network server 106, receives requests from NFS client 116 in network client 102. (*See*, col. 4, lines 31-37.) This request is passed to NFS server 122, which the Examiner has identified as Appellants' location server. Thus, whatever the cited passage means, it does not teach Appellants' name server receiving a file name and returning a file identifier.

b. Story Does Not Teach Appellants' Location Server

Claim 11 provides for a location server receiving the file identifier and returning file location information. The Examiner identified Story's NFS server 122 as Appellants' location server, with the following argument on page 26:

> Story discloses NFS server containing file set ID and file ID interfaced through element 126 as detailed in col 5, line 2-6, further Story's NFS server 122 is capable of both sending and receiving file information such as file identifier, location information and like because they are connected through network element 110 as detailed in fig 1.

The Examiner appears to present two arguments. The first argument appears to be that Story's file set ID and file ID correspond to Appellants' file identifier, received by NFS server 122 from interface 126. However, as can plainly be seen in Story's Figure 1, information only flows from NFS server 122 to interface 126. In addition, Story's Figure 2 illustrates that NFS server 122 receives a read/write request (step b) and forwards a read/write request to Story's

file system (step c). The text supporting Figure 2 confirms that Story's NFS server does not return file location information at column 5, lines 19-28 as follows (emphasis added):

> FIG. 2 shows an example of interaction between a network client and a network server. As shown, at step a, the application program is executed and sends a read or write system call to the NFS client. The NFS client *sends a READ or WRITE NFS request to the NFS server at step b*. This NFS request includes a file handle. A file handle, as previously described, is a data structure that contains sufficient information for uniquely identifying the file to be accessed. Then, *at step c, a READ or WRITE NFS request for a file is sent by the NFS server to the OSS file system.*

Story's NFS server does not, in any manner, generate and send file location information.

The Examiner's second argument appears to be that, since Story's NFS server is connected to a network, it sends location information through the network. There is no support for this conclusion. Further, Story's disclosure clearly indicates otherwise. Under the most generous interpretation of Figure 2, the only information that leaves the NFS server for the NFS client is the actual data in a read operation or the status in a write operation.

c. Story Does Not Teach a File Identifier Received From a Name Server and Sent to a Location Server

Claim 11 provides for receiving a file identifier corresponding to a file name from a name server and sending the file identifier to a location server. Thus, in order to support his claim that Story anticipates claim 11, the Examiner must find a file identifier that travels from Story's name server to Story's NFS server. The Examiner's response to Appellants' assertion that there is no such teaching in Story appears on page 27 as follows:

> As to the above argument, Examiner disagree with the applicant Story specifically teaches for example file identifier [see col 4, line 43-47, fig 4, element 404], more specifically file handler contains file identifier information, and all the requests would be processed through NFS server element 122 as detailed in col 4, line 44-52]

The passages cited by the Examiner are included in the text at column 4, lines 39-62, as follows:

> In a preferred embodiment of the invention, NFS server 122 may include multiple server processes for implementing multiple tasks. The NFS LAN interface process dispatches work to a server process in NFS server 122, based on the contents of a "file handle," which is a parameter in the NFS request. A file handle contains such information as the type of file, the time of creation of the file, a unique identifier (file set ID) for the file set in which the file resides, a unique identifier (file ID) for the file within the file set, etc. All requests for a given file set will go through the same NFS server process. NFS server 122, which receives an NFS request from the NFS LAN interface process, gains access to OSS file system 124 via a set of system calls through interface 126.
>
> OSS file system 124 supports disk files. It contains one or more file-system data structures called VNODEs (virtual nodes). Each file currently in use in the server has an associated VNODE. A VNODE contains various kinds of information about the state of the file, including whether it is open, where its cached data (if any) is located, the timestamps associated with the file, etc. A VNODE also contains an NFS "pseudo-open" status of the associated file. A pseudo-open describes the state of a file currently being accessed via the NFS server.

Figure 4 “shows a VNODE data structure ans a VNODE hash table used in a disk file system in the network server . . .” (Col. 3, ll. 44-45.) Reference 404 is to the “data structure of a VNODE.” (Col. 6, ll. 29-30.) Thus, Story teaches that NFS server 122 (supposedly Appellants’ location server) sends a file handle to the OSS file system, which represents each file with a VNDODE including the file state. There is no teaching that the file handle is sent from Story’s name server to Story’s NFS server or, in the event that the Examiner now means to claim that Story’s NFS server is Appellants’ name server, that Story’s VNODES somehow perform the function of Appellants’ location server. Perhaps the most telling indication that Story does not teach Appellants’ invention is that, in Story’s Figure 1, there is simply no information path that flows from Story’s name server 130 to Story’s NFS server 122. There

is no way that any information, let alone Appellants' file identifier, is received from Story's name server and is then sent to Story's NFS server.

The Examiner presents other arguments in response to Appellants' Brief. However, these arguments appear to be identical to those previously made by the Examiner and addressed in the Brief.

Appellants believe that claims 1-20, all claims pending in the application under appeal, are patentable. Appellants respectfully request that the Board so hold.

No fee is believed due by filing this paper. However, any fee due may be charged to Deposit Account No. 02-3978.

Respectfully submitted,

MARK A. BAKKE et al.

By: ______________________

Mark D. Chuey, Ph.D.
Registration No. 42,415
Agent for Appellant

Date: December 13, 2002

BROOKS & KUSHMAN P.C.
1000 Town Center, 22nd Floor
Southfield, MI 48075
Phone: 248-358-4400
Fax: 248-358-3351